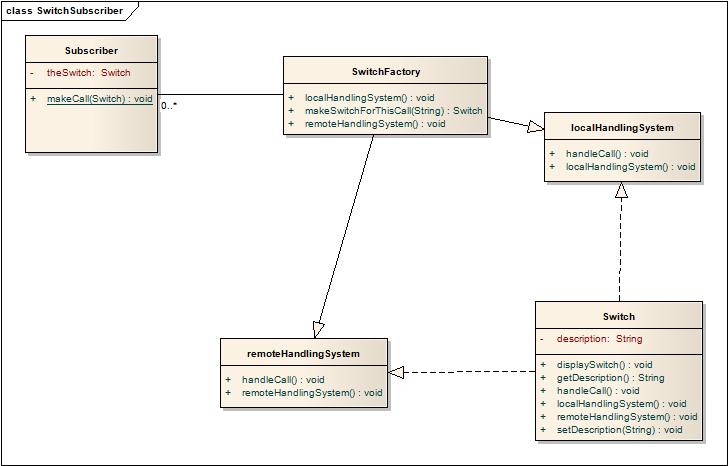
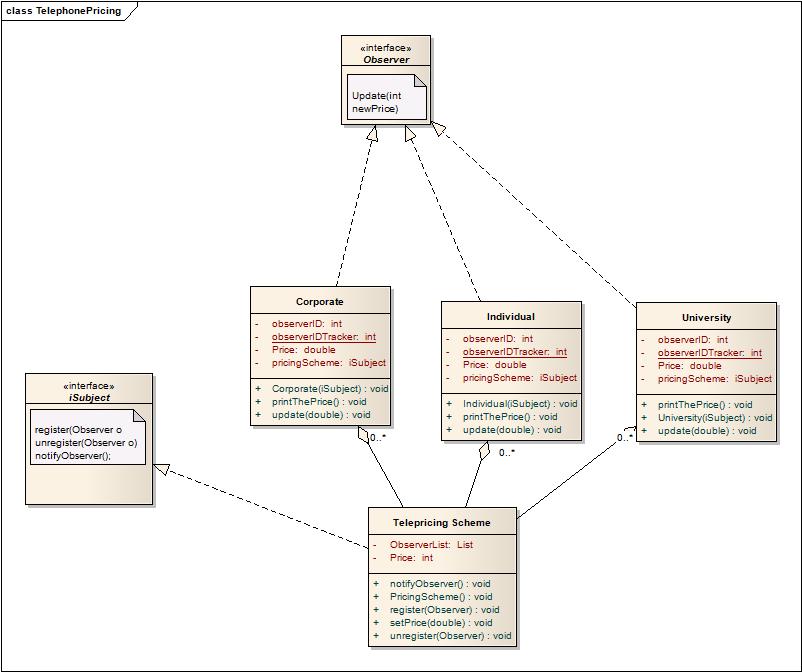
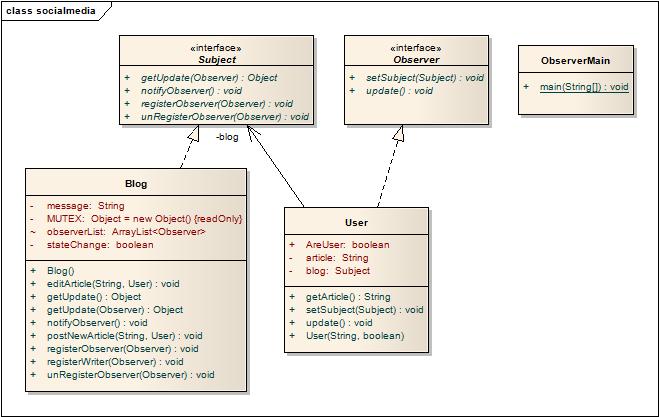
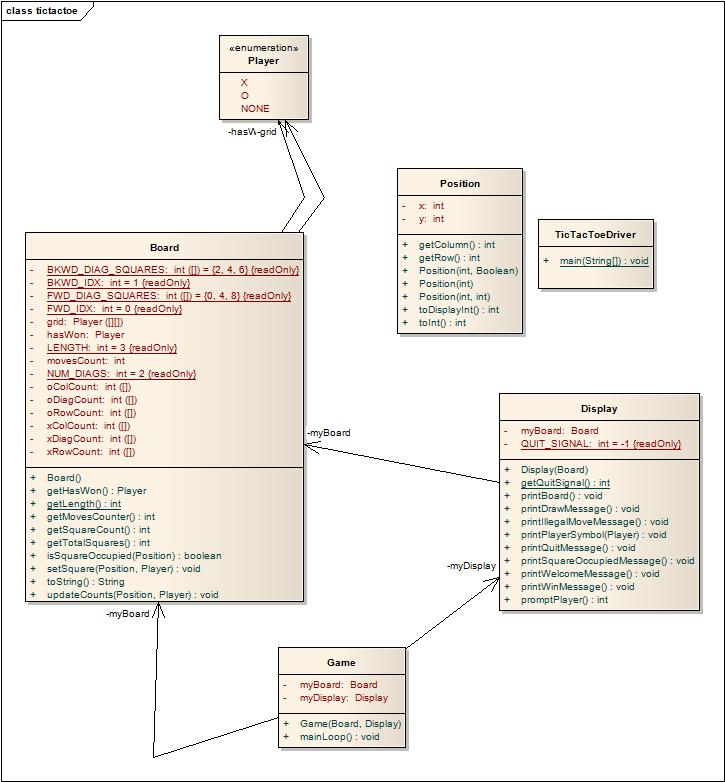
Cohort Exercise 3:

Cohort Exercise 5

Cohort Exercise 6



Homework Question 1:

Here, I used the MVC (Model, View Controller) design pattern. It decomposes the game into the model, the view and the controller subsystems.

The TicTacToeDriver class is the entry point of execution for the TicTacToe game. It instantiates new instances of the board, display and game classes representing the model, view and controller components respectively of the MVC design pattern.

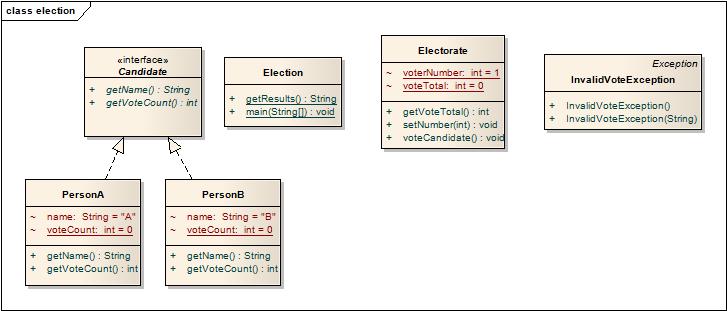
Position class represents a Cartesian coordinate indicating a square in the tic tac toe board Requires Board to exist in the same package to access static method Board.getLength().

Game class starts game and continues playing until a player has won, game is drawn or user requests to quit the game.

Board class represents Model of game. It initiates the 2D array representing the grid and checks if the player has won the game.

Display class represents View of the game. It receives input from the user and outputs messages and game state info to the console. This moves all user input/output to a single place (Display class) allowing other classes to deal with other functionality of the game.

Homework Question 2:



Here I used the Command design pattern. I created the *Candidate* interface, acting as a command. I have also created an *Electorate* class which acts as a request. I have Person A and Person B implementing *Candidate* interface which will do actual command processing.